

Towards Fully Automated TLM-to-RTL Property Refinement^{*}

Vladimir Herdt¹

Hoang M. Le¹

Daniel Große^{1,2}

Rolf Drechsler^{1,2}

¹Institute of Computer Science, University of Bremen, 28359 Bremen, Germany

²Cyber-Physical Systems, DFKI GmbH, 28359 Bremen, Germany

{vherdt,hle,grosse,drechsler}@informatik.uni-bremen.de

Abstract—An ESL design flow starts with a TLM description, which is thoroughly verified and then refined to a RTL description in subsequent steps. The properties used for TLM verification are refined alongside the TLM description to serve as starting point for RTL property checking. However, a manual transformation of properties from TLM to RTL is error prone and time consuming. Therefore, in this paper we propose a fully automated TLM-to-RTL property refinement based on a symbolic analysis of transactors. We demonstrate the applicability of our property refinement approach using a case study.

I. INTRODUCTION

In the recent years, the emergence of the *Electronic System Level* (ESL) [1] can be witnessed. It has become an industry common practice to model hardware designs at this new level of abstraction using the IEEE standard language SystemC [2], [3]. At ESL, functional behaviors of a design are described using the full descriptive power of C++ together with the added layer of concurrency by SystemC, while its communication interfaces are abstracted to function calls and parameters. The synchronization between functional behaviors and communication interfaces is realized using events. These modeling techniques are termed as *Transaction Level Modeling* (TLM) and standardized as a set of interfaces as TLM-2.0 [2].

In the top-down ESL design flow, SystemC TLM models are available very early and get successively refined to RTL. TLM models also serve as *reference models* to verify the corresponding RTL models later. The verification is commonly performed by a TLM/RTL co-simulation, where the same input stimuli are applied to both models and their outputs are checked for equivalence. This step requires an additional executable verification component, known as *transactor*, to bridge the function calls at TLM with the signal-based interfaces at RTL and vice versa, as shown in the upper half of Fig. 1.

Obviously, the correctness of TLM models is also of great importance. In the past few years, a wide body of verification techniques at TLM has been developed ranging from simulation-based (e.g. [4]–[6]) to formal verification (e.g. [7]–[14]). Please note that we do not try to be comprehensive but only mention a few representative approaches. Most of these approaches are based on the principle of *Assertion-Based Verification* (ABV) [15], which has been very successful for RTL verification. Assertions, also known as properties, provide a mean to formally capture the functional specification of the design. They specify conditions on design inputs, outputs and internal variables that are supposed to hold at all times. These

conditions can also be temporal, i.e. referring to a sequence of values over time. Assertions can be automatically translated into executable monitors to be co-simulated with the design to check the specified conditions. For the specification of TLM properties, extensions to standardized language such as IEEE-1850 PSL or IEEE-1800 SVA have been proposed [4], [16].

While the verification step based on TLM/RTL co-simulation is indispensable, the obtained results are far from complete and cannot ensure the absence of bugs at RTL. For some special cases when the TLM models are high-level synthesizable (e.g. when all used constructs are in the *SystemC Synthesizable Subset*), formal sequential equivalence checking is supported by existing EDA tools such as Calypto SLEC or Synopsys Hector. For general TLM models, it is highly desirable that co-simulation is complemented with other (preferably formal) approaches. A promising direction is to reuse properties that have been proven on the TLM model. Due to the semantic differences of the involved abstraction levels, straight-forward reuse is not possible. The translation process, i.e. TLM-to-RTL property refinement, is mostly manual, therefore error-prone and time-consuming.

Related Work: Several improvements to the manual TLM-to-RTL property refinement have been proposed. Ecker et al. [17] formulated a set of requirements for the refinement process. In a follow-up work [18], an automated refinement framework has been introduced. Still, a set of refinement rules have to be defined by verification engineers before the automated translation can start. Chen and Mishra [19] proposed a similar approach that requires a formal (temporal) semantic mapping between TLM functions and clocked RTL signals. Pierre and Amor [20] developed a set of pre-defined “pattern-matching” rules to ensure that the refined RTL properties belong to the *simple subset* of PSL, which is generally easier to verify. However, the need for a manual semantic mapping is still not circumvented. Bombieri et al. [21] proposed to reuse TLM properties in a TLM/RTL co-simulation. Thanks to the availability of transactors, RTL signals over time are converted back to TLM transactions and the TLM properties can thus be checked on the RTL design. This is, however, not a semantic approach, i.e. no corresponding RTL properties can be derived to apply, for example, RTL property checking.

Paper Contribution and Organization: To the best of our knowledge, we propose in this paper the first *fully automated* TLM-to-RTL property refinement approach. The lower half of Fig. 1 shows an overview. The main idea lies in a better reuse of the readily available transactors. At the core of our approach is a static transactor analysis based on symbolic execution (Section III). Essentially, the analysis reverse-engineers the executable transactors to create

^{*} This work was supported in part by the German Research Foundation (DFG) within the Reinhart Koselleck project DR 287/23-1, by the German Federal Ministry of Education and Research (BMBF) within the project CONFIRM under contract no. 16ES0565 and by the University of Bremen’s graduate school SyDe, funded by the German Excellence Initiative.

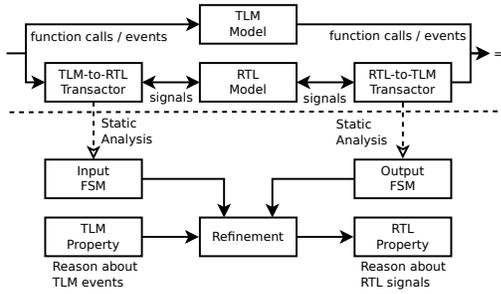


Fig. 1. TLM-to-RTL Property Refinement Overview

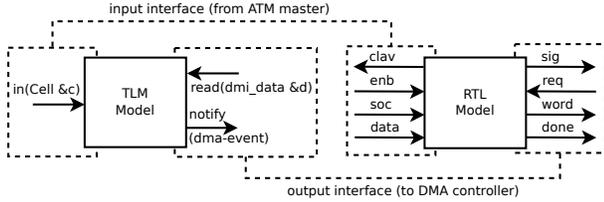


Fig. 2. UTOPIA controller TLM and RTL model IO interface (simplified)

a formal specification of the underlying protocol as *Finite State Machines* (FSM). Then, TLM properties are refined by relating high-level TLM events with RTL signal combinations at different clock cycles based on the FSM (Section IV). We believe that the proposed approach is of great practical interest. While it is possible to generate transactors automatically from a formal specification [22], in practice, verification engineers still develop transactors manually from scratch based on a textual specification. As the proposed approach can be best explained on concrete examples, we introduce in Section II a realistic case study for illustration and also feasibility demonstration.

II. UTOPIA CASE STUDY

As case study we consider an UTOPIA-based (*universal test and operations PHY interface for ATM*) controller acting as receiver slave device. It communicates with an *Asynchronous Transfer Mode* (ATM) master device using ATM cells, provides an internal buffer to store up to two cells, and a DMA interface to access the buffer. For illustrative purposes, we use a simplified data model, where a cell consists of 3 bytes and the DMA controller word size is 16 bits. DMA access is expected to be word-aligned thus the internal buffer cells are zero-padded (one byte here). Fig. 2 shows the TLM and RTL IO interface:

1) *TLM Interface*: The *in* function accepts a whole cell and writes it into the internal buffer. It blocks in case the buffer is full. The (dma-) read function provides a direct memory interface for reading a buffer cell (through a char pointer in *dmi_data*). The (dma-) event signalizes that a cell is available for reading (condition checked after every *in* or *read* call). For this interface, we have five TLM events for property specification: *in:begin*, *in:end*, *read:begin*, *read:end* and (dma-) *event:notified* (abbreviated *notified* as it is the only event).

2) *RTL Interface*: On the input side *clav* (cell available) signals the controller is able to receive a whole data cell, i.e. buffer not full. The master then starts the transfer by activating the *enb* (enable) signal, sets *soc* (start of cell) to 1 and data to the first byte of the cell. In subsequent clock cycles, the remaining cell bytes are transferred, one after another through *data* signal. During transfer, *soc* stays 0 and *enb* stays 1.

On the output side, *sig* (read signal) is active iff a whole cell is available in the internal buffer. DMA controller enables *req* (request) when ready to receive. Then, the data is provided sequentially through the *word* signal (2 byte steps). Finally, *done* is enabled one clock cycle after data transfer has finished.

3) *Transactors*: Fig. 3 shows the relevant implementation parts of the input transactor (TLM-to-RTL). Essentially, the transactor has two functions: a TLM input function to receive a cell and a RTL interface function to send it out. The TLM input function receives a cell and stores it internally. It will block (Line 8) in case a cell has already been received but not send out (*data_available=true*). The RTL interface function is sampled at every RTL clock cycle. It mimics the behavior of an ATM master device. Therefore, the transactor keeps track of the internal protocol status (Line 18). It starts in *status=WAIT* until the *clav* signal is observed. Then (*status=START*) it initiates data transfer once a TLM cell is available. In the next clock cycles (all *status=SEND*), the remaining data bytes are transferred (Line 41). The variable *i* tracks the next byte to send. Once the transfer is finished, the internal protocol data (*status* and *i*) are reset and the TLM input function is unblocked (Line 44-47), i.e. can receive the next cell. Asserts are used in the transactor to detect illegal signal combinations (e.g. *clav* is set to low before data is sent). The implementation of the output transactor (RTL-to-TLM) is similar to the input transactor, and therefore omitted.

III. STATIC ANALYSIS OF TRANSACTORS

This section describes our static analysis based on symbolic execution to extract the underlying transactor protocol between TLM and RTL as FSMs.

A. Symbolic Execution

Starting with the initial symbolic execution state, the transactor function is repeatedly executed, until all reachable states are explored. Fig. 4 shows the complete symbolic state space of the transactor. In total we have 7 distinct states ($\{S_0, \dots, S_6\}$), which make up the state space, and 6 additionally observed states ($\{*S_7, \dots, *S_{12}\}$) that are equivalent to one of the first 7 and thus are discarded.

The initial execution state S_0 is defined as follows: RTL signals and TLM input are externally provided and therefore initialized with symbolic values. The protocol data (*status* and index *i*) are initialized with concrete values. Beside the variable environment, a symbolic execution state has a path condition (*PC*, initialized to *True*, i.e. no constraints in S_0). Constraints are added when executing an *assert(C)*, i.e. $PC = PC \wedge C$, or branch with symbolic condition *C*, e.g. starting from S_0 execution will fork due to symbolic signal *clav* resulting in the states S_1 and S_6 (PC updated as $PC = PC \wedge C$ and $PC = PC \wedge \neg C$, respectively). Between subsequent executions protocol data is preserved, whereas the RTL signal values are reset to fresh symbolic values each time (as they can be modified in each clock cycle). The TLM input data is preserved between executions where the TLM input function cannot be called. This is the case for S_3 and S_4 , since *data_available* is set to *true*, which will block the input function. This information is collected by performing a preliminary static analysis on the input function. S_5 unblocks the input function again by setting *data_available* to *false* and notifying the awaited event *empty*. We annotate

```

1 struct TLMtoRTLtransactor {          14 }
2 bool data_available = false;        15 }
3 char data[CELL_SIZE];              16
4 event empty;                        17
5                                     18
6 void tlm_input(const Cell &c) {     19
7   if (data_available) {             20
8     wait(empty);                    21
9     assert(!data_available);        22
10  } else {                            23
11    data_available = true;           24
12    for (i=0; i<CELL_SIZE; ++i)     25
13      data[i] = c.data[i];          26
14                                     27
15                                     28
16                                     29
17                                     30
18                                     31
19                                     32
20                                     33
21                                     34
22                                     35
23                                     36
24                                     37
25                                     38
26                                     39
27                                     40
28                                     41
29                                     42
30                                     43
31                                     44
32                                     45
33                                     46
34                                     47
35                                     48
36                                     49
37                                     50
38                                     51
39                                     52
40                                     53
41                                     54
42                                     55
43                                     56
44                                     57
45                                     58
46                                     59
47                                     60
48                                     61
49                                     62
50                                     63
51                                     64
52                                     65
53                                     66
54                                     67
55                                     68
56                                     69
57                                     70
58                                     71
59                                     72
60                                     73
61                                     74
62                                     75
63                                     76
64                                     77
65                                     78
66                                     79
67                                     80
68                                     81
69                                     82
70                                     83
71                                     84
72                                     85
73                                     86
74                                     87
75                                     88
76                                     89
77                                     90
78                                     91
79                                     92
80                                     93
81                                     94
82                                     95
83                                     96
84                                     97
85                                     98
86                                     99
87                                     100
88                                     101
89                                     102
90                                     103
91                                     104
92                                     105
93                                     106
94                                     107
95                                     108
96                                     109
97                                     110
98                                     111
99                                     112
100                                    113
101                                    114
102                                    115
103                                    116
104                                    117
105                                    118
106                                    119
107                                    120
108                                    121
109                                    122
110                                    123
111                                    124
112                                    125
113                                    126
114                                    127
115                                    128
116                                    129
117                                    130
118                                    131
119                                    132
120                                    133
121                                    134
122                                    135
123                                    136
124                                    137
125                                    138
126                                    139
127                                    140
128                                    141
129                                    142
130                                    143
131                                    144
132                                    145
133                                    146
134                                    147
135                                    148
136                                    149
137                                    150
138                                    151
139                                    152
140                                    153
141                                    154
142                                    155
143                                    156
144                                    157
145                                    158
146                                    159
147                                    160
148                                    161
149                                    162
150                                    163
151                                    164
152                                    165
153                                    166
154                                    167
155                                    168
156                                    169
157                                    170
158                                    171
159                                    172
160                                    173
161                                    174
162                                    175
163                                    176
164                                    177
165                                    178
166                                    179
167                                    180
168                                    181
169                                    182
170                                    183
171                                    184
172                                    185
173                                    186
174                                    187
175                                    188
176                                    189
177                                    190
178                                    191
179                                    192
180                                    193
181                                    194
182                                    195
183                                    196
184                                    197
185                                    198
186                                    199
187                                    200
188                                    201
189                                    202
190                                    203
191                                    204
192                                    205
193                                    206
194                                    207
195                                    208
196                                    209
197                                    210
198                                    211
199                                    212
200                                    213
201                                    214
202                                    215
203                                    216
204                                    217
205                                    218
206                                    219
207                                    220
208                                    221
209                                    222
210                                    223
211                                    224
212                                    225
213                                    226
214                                    227
215                                    228
216                                    229
217                                    230
218                                    231
219                                    232
220                                    233
221                                    234
222                                    235
223                                    236
224                                    237
225                                    238
226                                    239
227                                    240
228                                    241
229                                    242
230                                    243
231                                    244
232                                    245
233                                    246
234                                    247
235                                    248
236                                    249
237                                    250
238                                    251
239                                    252
240                                    253
241                                    254
242                                    255
243                                    256
244                                    257
245                                    258
246                                    259
247                                    260
248                                    261
249                                    262
250                                    263
251                                    264
252                                    265
253                                    266
254                                    267
255                                    268
256                                    269
257                                    270
258                                    271
259                                    272
260                                    273
261                                    274
262                                    275
263                                    276
264                                    277
265                                    278
266                                    279
267                                    280
268                                    281
269                                    282
270                                    283
271                                    284
272                                    285
273                                    286
274                                    287
275                                    288
276                                    289
277                                    290
278                                    291
279                                    292
280                                    293
281                                    294
282                                    295
283                                    296
284                                    297
285                                    298
286                                    299
287                                    300
288                                    301
289                                    302
290                                    303
291                                    304
292                                    305
293                                    306
294                                    307
295                                    308
296                                    309
297                                    310
298                                    311
299                                    312
300                                    313
301                                    314
302                                    315
303                                    316
304                                    317
305                                    318
306                                    319
307                                    320
308                                    321
309                                    322
310                                    323
311                                    324
312                                    325
313                                    326
314                                    327
315                                    328
316                                    329
317                                    330
318                                    331
319                                    332
320                                    333
321                                    334
322                                    335
323                                    336
324                                    337
325                                    338
326                                    339
327                                    340
328                                    341
329                                    342
330                                    343
331                                    344
332                                    345
333                                    346
334                                    347
335                                    348
336                                    349
337                                    350
338                                    351
339                                    352
340                                    353
341                                    354
342                                    355
343                                    356
344                                    357
345                                    358
346                                    359
347                                    360
348                                    361
349                                    362
350                                    363
351                                    364
352                                    365
353                                    366
354                                    367
355                                    368
356                                    369
357                                    370
358                                    371
359                                    372
360                                    373
361                                    374
362                                    375
363                                    376
364                                    377
365                                    378
366                                    379
367                                    380
368                                    381
369                                    382
370                                    383
371                                    384
372                                    385
373                                    386
374                                    387
375                                    388
376                                    389
377                                    390
378                                    391
379                                    392
380                                    393
381                                    394
382                                    395
383                                    396
384                                    397
385                                    398
386                                    399
387                                    400
388                                    401
389                                    402
390                                    403
391                                    404
392                                    405
393                                    406
394                                    407
395                                    408
396                                    409
397                                    410
398                                    411
399                                    412
400                                    413
401                                    414
402                                    415
403                                    416
404                                    417
405                                    418
406                                    419
407                                    420
408                                    421
409                                    422
410                                    423
411                                    424
412                                    425
413                                    426
414                                    427
415                                    428
416                                    429
417                                    430
418                                    431
419                                    432
420                                    433
421                                    434
422                                    435
423                                    436
424                                    437
425                                    438
426                                    439
427                                    440
428                                    441
429                                    442
430                                    443
431                                    444
432                                    445
433                                    446
434                                    447
435                                    448
436                                    449
437                                    450
438                                    451
439                                    452
440                                    453
441                                    454
442                                    455
443                                    456
444                                    457
445                                    458
446                                    459
447                                    460
448                                    461
449                                    462
450                                    463
451                                    464
452                                    465
453                                    466
454                                    467
455                                    468
456                                    469
457                                    470
458                                    471
459                                    472
460                                    473
461                                    474
462                                    475
463                                    476
464                                    477
465                                    478
466                                    479
467                                    480
468                                    481
469                                    482
470                                    483
471                                    484
472                                    485
473                                    486
474                                    487
475                                    488
476                                    489
477                                    490
478                                    491
479                                    492
480                                    493
481                                    494
482                                    495
483                                    496
484                                    497
485                                    498
486                                    499
487                                    500
488                                    501
489                                    502
490                                    503
491                                    504
492                                    505
493                                    506
494                                    507
495                                    508
496                                    509
497                                    510
498                                    511
499                                    512
500                                    513
501                                    514
502                                    515
503                                    516
504                                    517
505                                    518
506                                    519
507                                    520
508                                    521
509                                    522
510                                    523
511                                    524
512                                    525
513                                    526
514                                    527
515                                    528
516                                    529
517                                    530
518                                    531
519                                    532
520                                    533
521                                    534
522                                    535
523                                    536
524                                    537
525                                    538
526                                    539
527                                    540
528                                    541
529                                    542
530                                    543
531                                    544
532                                    545
533                                    546
534                                    547
535                                    548
536                                    549
537                                    550
538                                    551
539                                    552
540                                    553
541                                    554
542                                    555
543                                    556
544                                    557
545                                    558
546                                    559
547                                    560
548                                    561
549                                    562
550                                    563
551                                    564
552                                    565
553                                    566
554                                    567
555                                    568
556                                    569
557                                    570
558                                    571
559                                    572
560                                    573
561                                    574
562                                    575
563                                    576
564                                    577
565                                    578
566                                    579
567                                    580
568                                    581
569                                    582
570                                    583
571                                    584
572                                    585
573                                    586
574                                    587
575                                    588
576                                    589
577                                    590
578                                    591
579                                    592
580                                    593
581                                    594
582                                    595
583                                    596
584                                    597
585                                    598
586                                    599
587                                    600
588                                    601
589                                    602
590                                    603
591                                    604
592                                    605
593                                    606
594                                    607
595                                    608
596                                    609
597                                    610
598                                    611
599                                    612
600                                    613
601                                    614
602                                    615
603                                    616
604                                    617
605                                    618
606                                    619
607                                    620
608                                    621
609                                    622
610                                    623
611                                    624
612                                    625
613                                    626
614                                    627
615                                    628
616                                    629
617                                    630
618                                    631
619                                    632
620                                    633
621                                    634
622                                    635
623                                    636
624                                    637
625                                    638
626                                    639
627                                    640
628                                    641
629                                    642
630                                    643
631                                    644
632                                    645
633                                    646
634                                    647
635                                    648
636                                    649
637                                    650
638                                    651
639                                    652
640                                    653
641                                    654
642                                    655
643                                    656
644                                    657
645                                    658
646                                    659
647                                    660
648                                    661
649                                    662
650                                    663
651                                    664
652                                    665
653                                    666
654                                    667
655                                    668
656                                    669
657                                    670
658                                    671
659                                    672
660                                    673
661                                    674
662                                    675
663                                    676
664                                    677
665                                    678
666                                    679
667                                    680
668                                    681
669                                    682
670                                    683
671                                    684
672                                    685
673                                    686
674                                    687
675                                    688
676                                    689
677                                    690
678                                    691
679                                    692
680                                    693
681                                    694
682                                    695
683                                    696
684                                    697
685                                    698
686                                    699
687                                    700
688                                    701
689                                    702
690                                    703
691                                    704
692                                    705
693                                    706
694                                    707
695                                    708
696                                    709
697                                    710
698                                    711
699                                    712
700                                    713
701                                    714
702                                    715
703                                    716
704                                    717
705                                    718
706                                    719
707                                    720
708                                    721
709                                    722
710                                    723
711                                    724
712                                    725
713                                    726
714                                    727
715                                    728
716                                    729
717                                    730
718                                    731
719                                    732
720                                    733
721                                    734
722                                    735
723                                    736
724                                    737
725                                    738
726                                    739
727                                    740
728                                    741
729                                    742
730                                    743
731                                    744
732                                    745
733                                    746
734                                    747
735                                    748
736                                    749
737                                    750
738                                    751
739                                    752
740                                    753
741                                    754
742                                    755
743                                    756
744                                    757
745                                    758
746                                    759
747                                    760
748                                    761
749                                    762
750                                    763
751                                    764
752                                    765
753                                    766
754                                    767
755                                    768
756                                    769
757                                    770
758                                    771
759                                    772
760                                    773
761                                    774
762                                    775
763                                    776
764                                    777
765                                    778
766                                    779
767                                    780
768                                    781
769                                    782
770                                    783
771                                    784
772                                    785
773                                    786
774                                    787
775                                    788
776                                    789
777                                    790
778                                    791
779                                    792
780                                    793
781                                    794
782                                    795
783                                    796
784                                    797
785                                    798
786                                    799
787                                    800
788                                    801
789                                    802
790                                    803
791                                    804
792                                    805
793                                    806
794                                    807
795                                    808
796                                    809
797                                    810
798                                    811
799                                    812
800                                    813
801                                    814
802                                    815
803                                    816
804                                    817
805                                    818
806                                    819
807                                    820
808                                    821
809                                    822
810                                    823
811                                    824
812                                    825
813                                    826
814                                    827
815                                    828
816                                    829
817                                    830
818                                    831
819                                    832
820                                    833
821                                    834
822                                    835
823                                    836
824                                    837
825                                    838
826                                    839
827                                    840
828                                    841
829                                    842
830                                    843
831                                    844
832                                    845
833                                    846
834                                    847
835                                    848
836                                    849
837                                    850
838                                    851
839                                    852
840                                    853
841                                    854
842                                    855
843                                    856
844                                    857
845                                    858
846                                    859
847                                    860
848                                    861
849                                    862
850                                    863
851                                    864
852                                    865
853                                    866
854                                    867
855                                    868
856                                    869
857                                    870
858                                    871
859                                    872
860                                    873
861                                    874
862                                    875
863                                    876
864                                    877
865                                    878
866                                    879
867                                    880
868                                    881
869                                    882
870                                    883
871                                    884
872                                    885
873                                    886
874                                    887
875                                    888
876                                    889
877                                    890
878                                    891
879                                    892
880                                    893
881                                    894
882                                    895
883                                    896
884                                    897
885                                    898
886                                    899
887                                    900
888                                    901
889                                    902
890                                    903
891                                    904
892                                    905
893                                    906
894                                    907
895                                    908
896                                    909
897                                    910
898                                    911
899                                    912
900                                    913
901                                    914
902                                    915
903                                    916
904                                    917
905                                    918
906                                    919
907                                    920
908                                    921
909                                    922
910                                    923
911                                    924
912                                    925
913                                    926
914                                    927
915                                    928
916                                    929
917                                    930
918                                    931
919                                    932
920                                    933
921                                    934
922                                    935
923                                    936
924                                    937
925                                    938
926                                    939
927                                    940
928                                    941
929                                    942
930                                    943
931                                    944
932                                    945
933                                    946
934                                    947
935                                    948
936                                    949
937                                    950
938                                    951
939                                    952
940                                    953
941                                    954
942                                    955
943                                    956
944                                    957
945                                    958
946                                    959
947                                    960
948                                    961
949                                    962
950                                    963
951                                    964
952                                    965
953                                    966
954                                    967
955                                    968
956                                    969
957                                    970
958                                    971
959                                    972
960                                    973
961                                    974
962                                    975
963                                    976
964                                    977
965                                    978
966                                    979
967                                    980
968                                    981
969                                    982
970                                    983
971                                    984
972                                    985
973                                    986
974                                    987
975                                    988
976                                    989
977                                    990
978                                    991
979                                    992
980                                    993
981                                    994
982                                    995
983                                    996
984                                    997
985                                    998
986                                    999
987                                    1000

```

Fig. 3. TLM-to-RTL transactor relevant implementation pseudocode

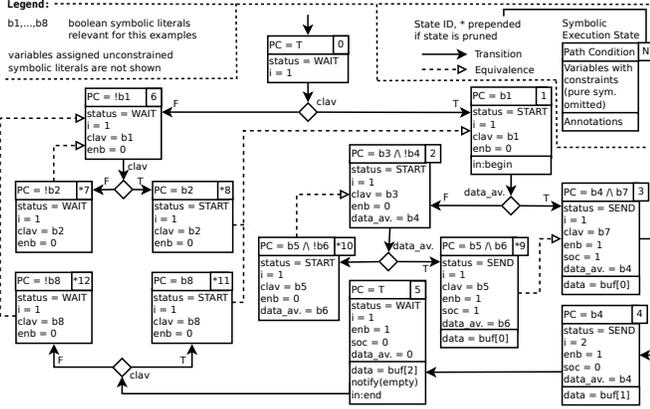


Fig. 4. Input transactor symbolic execution state space for the RTL interface function for the input transactor in Fig. 3, path conditions have been optimized by unused constraint elimination

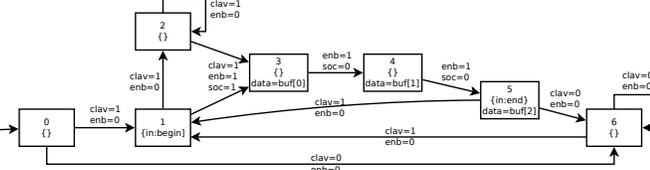


Fig. 5. FSM for the TLM-to-RTL transactor (input)

the input related TLM events, i.e. $in:b$ and $in:e$, to S1 and S5, respectively. The reason is that the execution of S1 is the first one to access input related data, and S5 the last one. Furthermore, we annotate the data mapping between the internal buffer buf and the RTL $data$ signal in S3, S4 and S5. Combining it with the data mapping of the input function, there exists a mapping between the RTL $data$ signal and TLM input cell at different states.

B. FSM Construction

The symbolic state space is transformed into an FSM by abstracting away all data except the RTL signal values. The FSM contains the same states and edges as the symbolic state space, but the edges are annotated with RTL signal values. For example starting from S0 either S1 (with $clav=1$ and $enb=0$) or S6 (with $clav=0$ and $enb=0$) can be reached. Please note, that the value of $clav$ is constrained to 1 (S1) and 0 (S6) due to the path condition. The other signal values are unconstrained and thus not mentioned. Signal combinations which are not covered, e.g. $clav=0$ and $enb=1$ in S0, are invalid based on the protocol. By repeating this process for all other edges, the transactor FSM is obtained. The input transactor FSM is shown in Fig. 5. We additionally preserve the TLM event annotations and data mappings in the FSM states. The output transactor FSM shown in Fig. 6 can be obtained similarly.

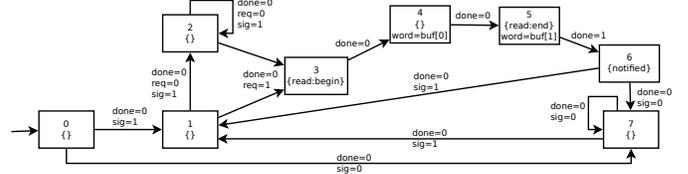


Fig. 6. FSM for the RTL-to-TLM transactor (output)

IV. PROPERTY REFINEMENT

This section outlines the refinement process based on the extracted FSMs. Before going into details, we briefly describe the used property language as well as the actual TLM properties for refinement.

A. Property Specification Language

For property specification, we consider the *simple subset* of PSL with TLM extensions similar to [4], [16]. For TLM properties, we use TLM events as atomic boolean propositions. For every function f we define the events $f:begin$ and $f:end$ (abbreviated as $f:b$ and $f:e$). Arguments can be captured to reason about the data, e.g. $in(x):e$ denotes that the in function has finished with x being passed as parameter. Finally, $e:notified$ denotes that event e has been notified. Boolean C++ expressions can be embedded to reason about data relations. We use SEREs (sequentialized regular expressions) to specify RTL traces (i.e. signal sequences sampled at clock cycles).

B. TLM Properties for Refinement

We consider two TLM properties P1 and P2, shown in upper part of Table I. P1 specifies that after a DMA read a DMA event notification needs to happen before the next DMA read is started. This ensures that data is available before it is attempted to read. P2 checks that no packets are lost. It captures the input packet in the variable x and when the input is completed, either the next or second next DMA read will read x . The $next_event(\alpha, \phi)$ operator requires that ϕ is true at the next occurrence of α . The $CEQ(y, x)$ predicate returns true when the DMA result and input cell are equal. The predicate can be expressed in C++ syntax as shown in Table I (upper part).

C. Refinement Process

Our refinement works recursively. The operators *always*, *next*, *before*, $|$, and $next_event(\alpha, \phi)$ are preserved. The implication $L \rightarrow R$ is mapped to $L | \rightarrow R$, i.e. first L needs to be matched and matching of R starts at the clock cycle L has been matched at. This is a valid refinement thanks to the monotonic advancement of time required by the simple subset.

A TLM event E is mapped to a sequence of RTL signals (i.e. traces) by following back all edges starting from FSM state S , event E is associated with, and collecting the signal assignments until the resulting traces T uniquely identify S ,

TABLE I
MAPPING OF TLM TO RTL PROPERTIES WITH REFINEMENT STEPS

CEQ(y,x)	$(y.\text{ptr}[0] \ \& \ 0\text{xff} == x.\text{data}[0]) \wedge (y.\text{ptr}[0] \gg 8 == x.\text{data}[1]) \wedge (y.\text{ptr}[1] \ \& \ 0\text{xff} == x.\text{data}[2])$
P1	always(read:e -> notify before read:b)
P2	always(in(x):e -> next_event(read(y):e, CEQ(y,x) next(next_event(read(z):e, CEQ(z,x)))))
RTL(notified)	done
_H1	(!done&&!sig done ; !done&&!sig) done ; !done&&sig ; !done&&req
_H2	(!done&&!sig done ; !done&&sig) !done&&!req&&sig ; !done&&!req&&sig
RTL(read:b)	_H1 _H2 ; !done&&req
RTL(read:e)	!done&&sig !done&&sig&&!req ; !done&&req ; !done ; !done
RTL(in(x):e)	(enb&&clav&&soc):NEW(x0:=data) ; (enb&&!soc):NEW(x1:=data) ; (enb&&!soc):NEW(x2:=data)
_S4	(done !done&&!sig ; !done&&sig) !done&&!req&&sig ; !done&&req ; !done
RTL(read(y):e)	_S4:NEW(y0:=word) ; !done:NEW(y1:=word)
RTL(CEQ(y,x))	$(y_0 \ \& \ 0\text{xff} == x_0) \wedge (y_0 \gg 8 == x_1) \wedge (y_1 \ \& \ 0\text{xff} == x_2)$
RTL(P1)	always(RTL(read:e) -> RTL(notify) before RTL(read:b))
RTL(P2)	always(RTL(in(x):e) -> next_event(RTL(read(y):e), RTL(CEQ(y,x))) next(next_event(RTL(read(z):e), RTL(CEQ(z,x)))))

Upper half shows the TLM properties and equality predicate CEQ, middle half shows mapping of TLM events to RTL signal sequences, and lower half shows the refined results.

i.e. starting from any state Q in the FSM and following any $t \in T$, only state S can be reached (otherwise there has been an invalid transition). We call T the distinguishing traces of state S , and can straightforwardly express T as a SERE by sequencing (operator $;$) all signal assignments within a trace and then union all traces (operator $|$). Table I (middle part) shows the results for the relevant TLM events. For example *notified* (associated with state $S6$ in output FSM) is uniquely identified by *done* (i.e. $done=1$), as $S5$ is the only state where *done* is a valid signal assignment. On the other hand to identify $S5$ (associated with *read:e*) going back one step is not enough, as many states can be reached with *!done*. In fact, T for $S5$ contains two traces of length 4.

Function call events capturing input/output arguments require an extended refinement procedure. It is not enough to just visit the FSM state associated with the function begin/end, but also go through all states where a data mapping to the arguments exists. The event $in(x):end$ requires to visit the state sequence $[S3, S4, S5]$ in the input FSM. Therefore, we generate T for the first state in the sequence and extend it with signal assignments to reach the subsequent states by following along the FSM edges. At each state we capture the current data signals in a new local variable for $S3$, e.g. $NEW(x_0:=data)$ (see [20]). Based on the data mapping from the FSMs, we obtain that RTL signal *data* in $S3$ (input FSM) is mapped to $buff[0]$ and following along the data mapping of the TLM input function, that $buff[0]$ maps to $c.data[0]$ of the input cell. Similarly, we obtain mappings for the output side. With the captured data signals and mappings, we can automatically refine the $CEQ(y,x)$ predicate (Table I lower part).

V. DISCUSSION AND FUTURE WORK

Our proposed *fully automated* TLM-to-RTL property refinement approach is still a work-in-progress. The monotonic advancement of time through the formula allows for an intuitive recursive refinement algorithm, where evaluation of subformulas starts at the clock cycle the previous one finished at. Although the feasibility of our refinement approach has been demonstrated on a realistic case study with two representative TLM properties, a formal characterization of supported properties is still to be defined in a future work. One limitation is that we assume the next (or SEREs in general) operator is not used to specify an absolute order of TLM events. The reason is, that it is unclear how to automatically refine it (especially when combined with other operators), as there can be (arbitrary) many clock cycles between TLM events at RTL. Moreover, it has been suggested that using the next operator in this way at TLM is not well suited (up to

the point of being unsound) due to its restrictiveness [20]. Rather, the *before* and *next_event* operators, which specify precedence, should be used. However, using a single next operator in combination with $->$ and *next_event* makes sense to avoid matching the subformula at the current instant (e.g. as used in our property P2 in Section IV-C to avoid matching the same *read*). This issue needs further investigation.

For property refinement, we compute distinguishing traces for FSM states. However, this is not possible for FSMs that have multiple state cycles, which accept the same input sequence. For a protocol description such an FSM implies that it is not possible to detect the protocol status by just observing the IO signals, i.e. there is no synchronization or handshaking in place. While we believe it is a reasonable assumption to have distinguishing states, our refinement algorithm can be extended to handle non-distinguishing states as well by: 1) accessing internal variables of the model, or 2) generating additional logic between model and monitor to simulate the FSM (can be done automatically). We plan to evaluate these extensions in future work.

REFERENCES

- B. Bailey, G. Martin, and A. Piziali, *ESL Design and Verification: A Prescription for Electronic System Level Methodology*. Morgan Kaufmann/Elsevier, 2007.
- IEEE Standard SystemC Language Reference Manual*, IEEE Std. 1666, 2011.
- D. Große and R. Drechsler, *Quality-Driven SystemC Design*. Springer, 2010.
- L. Ferro and L. Pierre, "Formal semantics for PSL modeling layer and application to the verification of transactional models," in *DATE*, 2010, pp. 1207–1212.
- D. Tabakov and M. Y. Vardi, "Monitoring temporal SystemC properties," in *MEMOCODE*, 2010, pp. 123–132.
- M. F. S. Oliveira et al., "The system verification methodology for advanced TLM verification," in *CODES+ISSS*, 2012, pp. 313–322.
- P. Herber, J. Fellmuth, and S. Glesner, "Model checking SystemC designs using timed automata," in *CODES+ISSS*, 2008, pp. 131–136.
- D. Große, H. M. Le, and R. Drechsler, "Proving transaction and system-level properties of untimed SystemC TLM designs," in *MEMOCODE*, 2010, pp. 113–122.
- A. Cimatti, I. Narasamy, and M. Roveri, "Software model checking SystemC," *TCAD*, vol. 32, no. 5, pp. 774–787, 2013.
- H. M. Le, D. Große, V. Herdt, and R. Drechsler, "Verifying SystemC using an intermediate verification language and symbolic simulation," in *DAC*, 2013, pp. 116:1–116:6.
- V. Herdt, H. M. Le, and R. Drechsler, "Verifying SystemC using stateful symbolic simulation," in *DAC*, 2015, pp. 49:1–49:6.
- V. Herdt, H. M. Le, D. Große, and R. Drechsler, "ParCoSS: efficient parallelized compiled symbolic simulation," in *CAV*, 2016, pp. 177–183.
- , "Compiled symbolic simulation for SystemC," in *ICCAD*, 2016, pp. 52:1–52:8.
- H. M. Le, V. Herdt, D. Große, and R. Drechsler, "Towards formal verification of real-world SystemC TLM peripheral models – a case study," in *DATE*, 2016, pp. 1160–1163.
- H. Foster, "Applied assertion-based verification: An industry perspective," *Found. Trends Electron. Des. Autom.*, vol. 3, no. 1, pp. 1–95, 2009.
- D. Tabakov, M. Vardi, G. Kamhi, and E. Singerman, "A temporal language for SystemC," in *FMCAD*, 2008, pp. 1–9.
- W. Ecker, V. Esen, T. Steininger, and M. Velten, "Requirements and concepts for transaction level assertion refinement," in *IESS*, 2007, pp. 1–14.
- T. Steininger, "Automated assertion transformation across multiple abstraction levels," Ph.D. dissertation, TU Munich, Germany, 2009.
- M. Chen and P. Mishra, "Assertion-based functional consistency checking between TLM and RTL models," in *VLSID*, 2013, pp. 320–325.
- L. Pierre and Z. B. H. Amor, "Automatic refinement of requirements for verification throughout the SoC design flow," in *CODES+ISSS*, 2013, pp. 29:1–29:10.
- N. Bombieri, F. Fummi, and G. Pravardelli, "Incremental abv for functional validation of tl-to-rtl design refinement," in *DATE*, 2007, pp. 882–887.
- F. Balarin and R. Passerone, "Specification, synthesis, and simulation of transactor processes," *TCAD*, vol. 26, no. 10, pp. 1749–1762, 2007.